EMERGING HEALTH TECHNOLOGIES

Role-Based Access Control (RBAC) Newsletter

Volume 33 – January 2007

Inside this issue:

VHA / IHS RBAC TF Chair
Robert O'Hara, MD
Robert.Ohara@med.va.gov

VHA Deputy Chief Architect
RBAC Project Manager
Steve Wagner
Steve.Wagner@med.va.gov

VHA Security Architect
RBAC Architect
Mike Davis, CISSP
Mike.Davis@med.va.gov

VHA Security Architect
RBAC Architect
Ed Coyne, PhD
Ed.Coyne@med.va.gov

VHA Security Analyst
Sepideh Khosravifar
Sepideh.Khosravifar@va.gov

RBAC Project Lead
Suzanne Webb
Suzanne.Gonzales-Webb@va.gov

Role-based Authorization Constraints Specification Using Object Constraint Language

Gail-Joon Ahn

Department of Computer Science, University of North Carolina at Charlotte

Michael. E. Shin

Department of Information and Software Engineering, George Mason University

Abstract

Per Ahn and Shin, the purpose of access control is to limit the actions on a computer system that a legitimate user can perform. Role-based access control or "RBAC," has generated great interest in the security community as a flexible approach in access control. One of the important aspects in RBAC is constraints that constrain what components in RBAC are allowed to do. Although researchers have identified useful constraints using formal specification languages such as RCL2000, there still exists per Ahn and Shin a demand to have constraints specification languages for system developers who are working on secure systems development. In this paper Ahn and Shin discuss another approach to specify constraints using a de facto constraints specification language in the software engineering arena. Ahn and Shin mention use of "Object Constraint Language (OCL)" that is a declarative language which is part of the Unified Modeling Language (UML) and has been used in object-oriented analysis and design. A description of how to specify previously identified role-based authorization constraints and future direction of this work is also addressed.

Introduction

The role-based access control (RBAC) is a flexible approach that has generated great interest in the security community [1]. RBAC has emerged as a widely accepted alternative to classical discretionary and mandatory access controls [2]. RBAC regulates the access of users to information and system resources based on activities that users need to execute in the system. It requires the identification of roles in the system wherein a role can be defined as a set of actions and responsibilities associate with a particular working activity. Then, instead of specifying all the accesses each individual user is allowed, access authorizations on objects are specified for roles. Since roles in an organization are relatively persistent with respect to user turnover and task re-assignment, RBAC provides a powerful mechanism for reducing the complexity, cost, and potential for error in assigning permissions to users within the organization. Because roles within an organization typically have overlapping permissions, RBAC models include features to establish role

Upcoming Meetings

- **HL7 January Working Group Meeting**
  January 7-12, 2007
  San Diego, CA
- **American Health Information Confidentiality, Privacy, and Security Workgroup:**
  January 8, 2007
  'listen in' at
  http://www.hhs.gov/healthit/ahic/cps_instruct.html
- **INCITS Meetings**
  January 23-25, 2007
  Houston, TX
- **INCITS Technical Committee CS1, Cyber Security**
  January 24-25, 2006
  San Jose, CA
- **First ACM Workshop on Education in Computer Security**
  January 29-31, 2007
  Monterey, CA
- **HIMSS**
  February 25-March 1, 2007
  New Orleans, LA

≈

hierarchies, where a given role can include all of the permissions of another role.

In this paper Ahn and Shin's focus is on constraints specification, i.e., how constraints can be expressed. Constraints can be expressed in natural languages, such as English, or in more formal languages. Natural language specification has the advantage of ease of comprehension by human beings, but may be prone to ambiguities.

The constraints in RBAC may be one of the most important components that enforce the principal motivations of the RBAC model. Authorization constraints (also simply called constraints) are another fundamental aspect of RBAC. Although the importance of constraints in RBAC has been recognized for a long time, they have not received much attention in the research literature, while role hierarchies have been practiced and discussed at considerable length.

Recently Ahn and Sandhu [3] proposed a formal language called RCL2000 (Role-based constraints specification language 2000) and identified useful role-based authorization constraints such as *prohibition* and *obligation* constraints. The users of RCL2000 are security researchers and security policy designers who have to understand organizational objectives and articulate major policy decisions to support theses objectives. RCL2000 also provides *n*-ary expressions and more flexibility, sharing a great deal of common semantics about expressing access control constraints [4].

Using OCL that has been used to express constraints in analysis and design as an industrial standard constraints specification language, Ahn and Shin demonstrate that OCL can help the reader specify previously identified constraints at the system design step. The constraints include separation of duty constraints, prerequisite constraints, and cardinality constraints. This approach is comparatively convenient for system developers to specify and to understand constraints of the RBAC model.

Related Technologies, Role-based Access Control

RBAC has recently received considerable attention as a promising alternative to traditional discretionary (DAC) and mandatory (MAC) access controls (see, for example, [2,5,6,7]). MAC is used in the classical defense arena; the policy of access is based on the classification of objects such as top-secret level. The main idea of DAC is that the owner of an object has discretionary authority over who else can access that object. RBAC policy is based on the roles of the subjects and can specify security policy in a way that maps to an organization's structure. A general family of RBAC models called RBAC96 was defined by Sandhu et al [2]; Figure 1 (below) illustrates the most general model in this family.

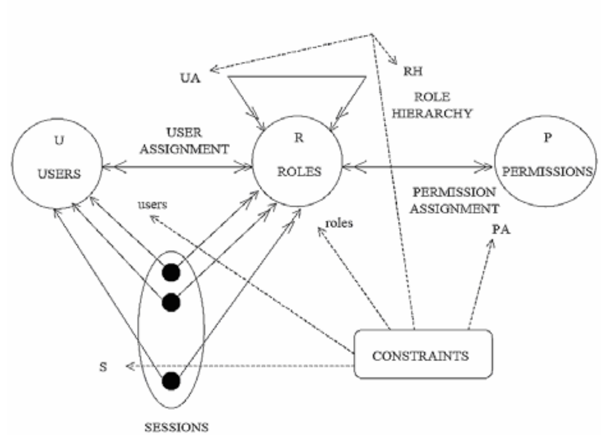
Motivation and discussion about various design decisions made in developing this family of models is given in [2]. Figure 1 shows (regular) roles and permissions that regulate access to data and resources.

Upcoming Meetings

- **INCITS CS1.1 Meeting**
  March 6-8, 2007
  Silicon Valley, CA
- **3rd ACM Conference on Computer & Communications Security**
  March 14-15, 2007
  New Delhi, India
- **4th ACM Conference on Computer & Communications Security**
  April 1-4, 2007
  Zurich, Switzerland
- **INCITS Meetings**
  April 10-12, 2007
  Redmond, WA
- **OASIS Symposium: eBusiness and Open Standards**
  April 15-20, 2007
  San Diego, CA
- **HL7 May Working Group Meeting**
  April 29-May 4, 2007
  Cologne, Germany
- **Web Services Security Conference & Exhibition**
  May 8-9, 2007
  Baltimore, MD
- **2007 IEEE Symposium on Security and Privacy**
  May 20-23, 2007
  Berkeley/Oakland , CA

≈

Figure 1 RBAC Model



Accordingly, Ahn and Shin state in their paper that a user can be a member of many roles and a role can have many users. Similarly, a role can have many permissions and the same permissions can be assigned to many roles. Each session relates one user to possibly many roles. Intuitively, a user establishes a session during which the user activates some subset of roles that he or she is a member of. The permissions available to the users are the union of permissions from all roles activates in that session. Each session is associated with a single user. This association remains constant for the life of a session. A user is a human being or an autonomous agent, a role is a job function or a job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role, and a permission is an approval of a particular mode of access to one or more objects in the system or some privilege to carry out specified actions.

A user may have multiple sessions open at the same time, each in a different window on the workstation screen for instance. Each session may have a different combination of active roles. The concept of a session equates to the traditional notation of a subject in access control. A subject is a unit of access control, and a user may have multiple subjects (or sessions) with different permissions active at the same time. There is a collection of constraints that allow or forbid values of various components of the RBAC model.

Roles are organized in a partial order $\geq$, so that if $x \geq y$ then role x inherits the permissions of role y. Members of x are also implicitly members of y. In such cases, [Ahn/Shin] say x is senior to y. Each session relates one user to possibly many roles. The idea is that a user establishes a session and activates some subset of roles that he or she is a member of (directly or indirectly by means of the role hierarchy). The RBAC model has the following components and these components are formalized from the above discussions.

- U is a set of users,
- R is disjoint sets of roles and administrative roles respectively,

Upcoming Meetings

- **SACMAT '01: 6th ACM Symposium on Access Control Models and Technologies (formerly known as RBAC)**
  May 3-4, 2007
  Chantilly, VA
- **SACMAT '02: 7th ACM Symposium on Access Control Models and Technologies**
  June 3-4, 2007
  Monterey, CA
- **SACMAT '03: 8th ACM Symposium on Access Control Models and Technologies 2003**
  June 1-4, 2007
  Como, Italy
- **Computer Security Institute (CSI) NetSec '07**
  June 11-13, 2007
  Scottsdale, AZ
- **12th ACM SACMAT**
  June 20-22, 2007
  Sophia Antipolis, France
- **6th Annual Security Conference**
  April 11-12, 2007
  Las Vegas, NV
  http://www.security-conference.org/

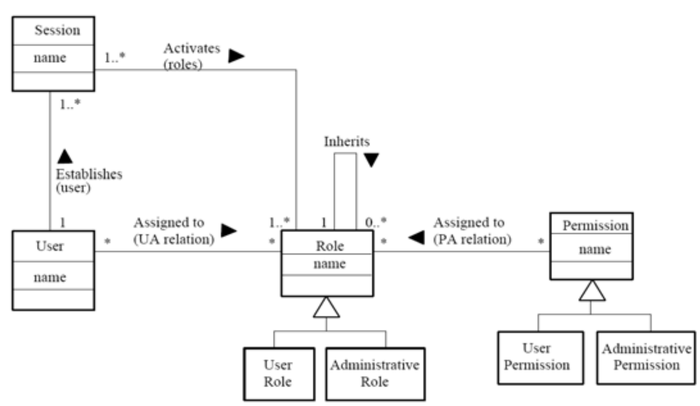
≈

- *P* is disjoint sets of permissions and administrative permissions,
- *UA Í U ´ R*, is a many-to-many user to role assignment relation,
- *PA Í P ´ R* is a many-to-many permission to role assignment relation,
- *RH Í R ´ R* is partially ordered role hierarchies (written as ≥ in infix notation),
- *S* is a set of sessions,
- *user: S ® U*, is a function mapping each session *si* to the single user *user(si)* and is constant for the session's lifetime, and
- *roles: S ®* 2*R* is a function mapping each session *si* to a set of roles *roles(si)* ⊆ {*r* | (∃*r'* ≥ *r*) [(*user*(*si*), *r'*) ∈ *UA*]} (which can change with time) so that session *si* has the permissions U*r*∈*roles* (*si*) {*p* | (∃*r²* ≤ *r*) [(*p*, *r²*) ∈ *PA*]}.

Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a general purpose visual modeling language in which we can specify, visualize, and document the components of software systems. The UML has become a standard modeling language in the field of software engineering.

Fig.2: Conceptual Class Model for RBAC - entity classes



The UML consists of functional, static, and dynamic models. In a functional model, the functional requirements of systems are specified using use case diagrams. A use case defines the services that a system provides to users. A static model provides a structural view of information in a system. Classes are defined in terms of their attributes and relationships. The relationships include association, generalization/specialization, and aggregation of classes. A dynamic model shows a behavioral view of a system. It can be described with collaboration diagrams, sequence diagrams, and statechart diagrams. A collaboration diagram and sequence diagram are developed to capture how objects collaborate with each other to execute a use case. State dependent views of objects are defined in statechart diagrams.

Object Constraint Language

The Object Constraint Language (OCL) [8, 9] is an expression language that describes constraints on object-oriented models. As defined for this paper, a constraint is a restriction on one or more values of an object-oriented model, wherein OCL is an industrial standard for object-oriented analysis and design.

Role-based Constraints

Constraints are an important aspect of access control and are a powerful mechanism for laying out a higher-level organizational policy. Consequently the specification of constraints needs to be considered. This issue has received surprisingly little attention in the research literature. There is some work such as [12,13] that deal with constraints in the context of role-based access control. This work, however, is preliminary and tentative, and needs substantial further development.

Chen and Sandhu [12] suggested how constraints could be specified. Giuri and Iglio [13] defined a new model to provide the capability of defining constraints on roles.

Constraints Specification

The conceptual static model for RBAC is depicted in Figure 2. It contains classes, their attributes, and their relationships [14]. The basic entities are user, role, permission, constraint, and session classes. The role can be specialized to user and administrative roles. The permission can also be specialized to user and administrative permissions. Each class has an attribute, that is, a name, which can be an identification of instance of the class. In the class model, the UA and PA relations indicate that users can be assigned to roles and permissions can be assigned to roles, respectively. Next, is the need to express constraints that regulate the construction and the activities of each class from this UML representation. Our expression includes separation of duty constraints, prerequisite constraints, and cardinality constraints.

1. **Separation of duty constraints**

Separation of duty is a well-known principle for preventing fraud by identifying conflicting roles—such as Purchase Manager and Accounts Payable Manager—and ensuring that the same individual can belong to at most one conflicting role. We may apply this conflicting notion to other components such as user and permission in role-based access control. The concept of conflicting permissions defines conflict in terms of permissions rather than roles. Thus the permission to issue purchase orders and the permission to issue payments are conflicting, irrespective of the roles to which they are assigned. Conflict defined in terms of roles allows conflicting permissions to be assigned to the same role by error (or malice). Conflict defined in terms of permissions eliminates this possibility.

2. **Prerequisite constraints**

This constraint is based on the concept of prerequisite roles introduced in [2]. For example, a user can be assigned to the engineer role only if the user already is assigned to the employee role. It ensures that only users who are already assigned to the employee role can be assigned to the engineer role. We call this kind of constraint as prerequisite-role constraints. The following examples demonstrate that OCL can also specify prerequisite constraints.

RBAC Newsletter

ATTN: Suzanne Webb
RBAC Project Lead
10260 Campus Point MS-B1E
La Jolla, CA 92121

Or e-mail:

Suzanne.Gonzales-Webb@va.gov

≈

3. Cardinality constraints

Another constraint type is a numerical limitation for classes in a role-based system. When applying cardinality as a constraint for example, in a healthcare setting only one user would have the access ability to hold the permissions of a Head Nurse on a hospital ward floor at any given time. This numerical limitation may vary depending upon the organizational policy. This shows per [Ahn, Shin] that OCL can specify these constraints without any extension of language.

Conclusion

In their paper, [Ahn/Shin] demonstrated that role-based authorization constraints can be specified using the industrial standard constraints specification language, OCL. [Ahn/Shin] have specified separation of duty constraints, prerequisite constraints and cardinality constraints. As a result, [Ahn/Shin] can utilize constraints identified by a formal language such as RCL2000 when we design and analyze role-based systems. [Ahn/Shin] believe that this work helps system developer understand constraints and requirements on secure systems development. There is room for much additional work with our approach. Validation of OCL specifications and time-based constraints can be studied. A unified way to specify authorization constraints can be investigated so that we can apply our approach to other access control models such as MAC and DAC.

References

[1] James Joshi, Arif Ghafoor, Walid Aref, and Eugene Spafford, "Digital Government security infrastructure design challenges," IEEE Computer, Volume 34, Number 2, pages 66-72, February 2001.

[2] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman, "Role-based access control models," IEEE Computer, Volume 29, Number2, pages 38-47, February 1996.

[3] Gail-Joon Ahn and Ravi Sandhu, "Role-based Authorization Constraints Specification," ACM Transactions on Information and Systems Security, Volume 3, Number 4, November 2000.

[4] Trent Jaeger and Jonathon Tidswell, "Practical Safety in Flexible Access Control Models." ACM Transactions on Information and Systems Security, Volume 4, Number 3, August 2001, to appear.

[5] David Ferraiolo and Richard Kuhn, "Role based access controls," In Proceedings of 15th NISTNCSC National Computer Security Conference, pages 554-563, Baltimore, MD, October 1316 1992.

[6] M.Y. Hu, S.A. Demurjian, and T.C. Ting, "User-role based security in the ADAM object-oriented design and analyses environment," In J. Biskup, M. Morgernstern, and C. Landwehr, editors, Database Security VIII: Status and Prospects, North-Holland, 1995.

RBAC Newsletter

ATTN: Suzanne Webb
RBAC Project Lead
10260 Campus Point MS-B1E
La Jolla, CA 92121

Or e-mail:

Suzanne.Gonzales-Webb@va.gov

≈

[7] Imtiaz Mohammed and David M. Dilts, "Design for dynamic user-role-based security," Computers & Security, Volume 13, Number 8, pages 661-671, 1994.

[8] OMG Web site, Unified Modeling Language Notation Guide, Version 1.3, September 2000.

[9] Jos Warmer and Anneke Kleppe, "The Object Constraint Language: Precise Modeling with UML," Addison-Wesley, 1999

[12] Chen, F. and Sandhu, R., "Constraints for role based access control," In Proceedings of the 1st ACM Workshop on Role-Based Access Control, pages 39-46, Gaithersburg, MD, November 30-December 1 1995.

[13] Giuri, L. and Iglio, P., "A formal model for role-based access control with constraints," In Proceedings of 9th IEEE Computer Security Foundations Workshop, pages 136-145, Kenmare, Ireland, June 1996.

[14] Michael E. Shin and Gail-Joon Ahn, "UML-based Representation of Role-Based Access Control," Fifth International Workshop on Enterprise Security (WETICE 2000), Gaithersburg, MD, June, 2000

RBAC Taskforce – Update

The next RBAC Taskforce meeting call will be held the first Wednesday of the month (February 7; March 7; April 4, 2007) at 1300CT / 1100PST / 1200MT / 1400EST; a meeting reminder will be sent to current participants. If you would like to be a part of the Task Force please contact Suzanne Gonzales-Webb for more information, thank you.

The RBAC Taskforce will continue discussions surrounding the definition of constraints on current Permission Catalog and Roles, as well as an update on the initiation of RBAC incorporation into the VA re-engineering projects. Current Task Force Members are contacted with additional materials in preparation for the meeting.

≈

The latest RBAC Documentation additions and prior RBAC Newsletters can be found on the RBAC Website.

≈

The RBAC Newsletter is a quarterly publication of the VHA RBAC Task Force. Please be on the lookout for the next issue due April 2007!

RBAC Newsletter

ATTN: Suzanne Webb
RBAC Project Lead
10260 Campus Point MS-B1E
La Jolla, CA 92121

Or e-mail:

Suzanne.Gonzales-Webb@va.gov

≈